SMT-based Synthesis of Safe and Robust PID Controllers for Stochastic Hybrid Systems

Fedor Shmarov[1], Nicola Paoletti[2], Ezio Bartocci[3], Shan Lin[4], Scott A. Smolka[2], and Paolo Zuliani[1]

[1] School of Computing, Newcastle University, UK
{f.shmarov,paolo.zuliani}@ncl.ac.uk
[2] Department of Computer Science, Stony Brook University, NY, USA
{nicola.paoletti,sas}@cs.stonybrook.edu
[3] Faculty of Informatics, TU Wien, Austria
ezio.bartocci@tuwien.ac.at
[4] Department of Electrical and Computer Engineering, Stony Brook University, NY, USA
shan.x.lin@stonybrook.edu

Abstract. We present a new method for the automated synthesis of safe and robust Proportional-Integral-Derivative (PID) controllers for stochastic hybrid systems. Despite their widespread use in industry, no automated method currently exists for deriving a PID controller (or any other type of controller, for that matter) with safety and performance guarantees for such a general class of systems. In particular, we consider hybrid systems with nonlinear dynamics (Lipschitz-continuous ordinary differential equations) and random parameters, and we synthesize PID controllers such that the resulting closed-loop systems satisfy safety and performance constraints given as probabilistic bounded reachability properties. Our technique leverages SMT solvers over the reals and nonlinear differential equations to provide formal guarantees that the synthesized controllers satisfy such properties. These controllers are also robust by design since they minimize the probability of reaching an unsafe state in the presence of random disturbances. We apply our approach to the problem of insulin regulation for type 1 diabetes, synthesizing controllers with robust responses to large random meal disturbances, thereby enabling them to maintain blood glucose levels within healthy, safe ranges.

1 Introduction

Proportional-Integrative-Derivative (PID) controllers are among the most widely deployed and well-established feedback-control techniques. Application areas are diverse and include industrial control systems, flight controllers, robotic manipulators, and medical devices. The PID controller synthesis problem entails finding the values of its control parameters (proportional, integral and derivative gains) that are optimal in terms of providing stable feedback control to the target system (the plant) with desired response behavior. Despite the limited number of parameters, this problem is far from trivial, due to the presence of multiple (and often conflicting) performance criteria that a controller is required to meet (*e.g.*, normal transient response, stability).

Developing PID controllers for *cyber-physical systems* is even more challenging because their dynamics are typically hybrid, nonlinear, and stochastic in nature. Moreover, it is imperative that the closed-loop controller-plus-plant system is safe (*i.e.*, does not reach a bad state) and robust (*i.e.*, exhibits desired behavior under a given range of disturbances). To the best of our knowledge, however, the current techniques for synthesizing PID controllers (see *e.g.*, [33,9,13]) simply ignore these issues and do not provide any formal guarantees about the resulting closed-loop system.

In this paper, we present a new framework for the automated synthesis of PID controllers for *stochastic hybrid systems* such that the resulting closed-loop system *provably* satisfies a given (probabilistic) safety property in a robust way with respect to random disturbances. Specifically, we formulate and tackle two different, yet complementary, problems: *controller synthesis*, *i.e.*, find a PID controller that minimizes the probability of violating the property, thus ensuring robustness against random perturbations; and *maximum disturbance synthesis*, *i.e.*, find, for a given controller, the largest disturbance that the resulting control system can sustain without violating the property. To the best of our knowledge, we are the first to present a solution to these problems (see also the related work in Section 6) with formal guarantees.

It is well known that safety verification is an inherently difficult problem for nonlinear hybrid systems — it is in general undecidable, hence it must be solved using approximation methods. Our technique builds on the frameworks of delta-satisfiability [16] and probabilistic delta-reachability [31] to reason formally about nonlinear and stochastic dynamics. This enables us to circumvent undecidability issues by returning solutions with numerical guarantees up to an arbitrary user-defined precision.

We express safety and performance constraints as probabilistic bounded reachability properties, and encode the synthesis problems as SMT formulae over ordinary differential equations. This theory adequately captures, besides the reachability properties, the hybrid nonlinear dynamics that we need to reproduce, and leverages appropriate SMT solvers [17,30] that can solve the delta-satisfiability problem for such formulae.

We demonstrate the utility of our approach on an artificial pancreas case study, *i.e.* the closed-loop insulin regulation for type 1 diabetes. In particular, we synthesize controllers that can provide robust responses to large random meal disturbances, while keeping the blood glucose level within healthy, safe ranges.

To summarize, in this paper, we make the following main contributions:

- We provide a solution to the *PID controller synthesis* and *maximum disturbance synthesis* problems using an SMT-based framework that supports hybrid plants with *nonlinear ODEs* and *random parameters*.
- We encode in the framework safety and performance requirements, and state the corresponding formal guarantees for the *automatically synthesized* PID controllers.
- We demonstrate the practical utility of our approach by synthesizing provably safe and robust controllers for an artificial pancreas model.

2 Background

Hybrid systems extend finite-state automata by introducing continuous state spaces and continuous-time dynamics [2]. They are especially useful when modeling systems that

combine discrete and continuous behavior such as cyber-physical systems, including biomedical devices (*e.g.*, infusion pumps and pacemakers). In particular, continuous dynamics is usually expressed via (solutions of) ordinary differential equations (ODEs). To capture a wider and more realistic family of systems, in this work we consider hybrid systems whose behavior depends on both *random* and *nondeterministic* parameters, dubbed *stochastic parametric hybrid systems (SPHS)* [31]. In particular, our synthesis approach models both the target system and its controller as a single SPHS. It is thus important to adopt a formalism that allows random *and* nondeterministic parameters: the former are used to model system disturbances and plant uncertainties, while the latter are used to constrain the search space for the controller synthesis.

Definition 1. (SPHS)*[31] A Stochastic Parametric Hybrid System is a tuple* $H =< Q, \Upsilon, X, P, Y, R, \mathsf{jump}, \mathsf{goal} >$*, where*

- $Q = \{q_0, \cdots, q_m\}$ *is the set of modes (discrete states) of the system;*
- $\Upsilon \subseteq \{(q,q') : q,q' \in Q\}$ *is the set of possible mode transitions (discrete dynamics);*
- $X = [u_1, v_1] \times \cdots \times [u_n, v_n] \times [0,T] \subset \mathbb{R}^{n+1}$ *is the continuous system state space;*
- $P = [a_1, b_1] \times \cdots \times [a_k, b_k] \subset \mathbb{R}^k$ *is the parameter space of the system, which is represented as* $P = P_R \times P_N$*, where* P_R *is domain of random parameters and* P_N *is the domain of nondeterministic parameters (and either domain may be empty);*
- $Y = \{\mathbf{y}_q(\mathbf{p}) : q \in Q, \mathbf{p} \in X \times P\}$ *is the continuous dynamics where* $\mathbf{y}_q : X \times P \to X$*;*
- $R = \{\mathbf{g}_{(q,q')}(\mathbf{p}) : (q,q') \in \Upsilon, \mathbf{p} \in X \times P\}$ *is the set of 'reset' functions* $\mathbf{g}_{(q,q')} : X \times P \to X \times P$ *defining the continuous state at time* $t = 0$ *in mode* q' *after taking the transition from mode* q*.*

and predicates (or relations)

- $\mathsf{jump}_{(q,q')}(\mathbf{p})$ *is true iff the discrete transition* $(q,q') \in \Upsilon$ *may occur upon reaching state* $(\mathbf{p}, q) \in X \times P \times Q$*,*
- $\mathsf{goal}_q(\mathbf{p})$ *is true iff* $\mathbf{p} \in X \times P$ *is a goal state for mode* q*.*

The goal predicate is the same for all modes and is used to define the safety requirements for the controller synthesis (see (4.6) in Section 4). We assume that the SPHS has an initial state $(\mathbf{x_0}, q_0) \in X \times Q$. The continuous dynamics Y is given as an initial-value problem with Lipschitz-continuous ODEs over a bounded time domain $[0,T]$, which have a unique solution for any given initial condition $\mathbf{p} \in X \times P$ (by the Picard-Lindelöf theorem). System parameters are treated as variables with zero derivative, and thus are part of the initial conditions. Finally, parameters may be random discrete/continuous (capturing system disturbances and uncertainties) with an associated probability measure, and/or nondeterministic (*i.e.* the parameters to synthesize), in which case only their bounded domain is known.

Probabilistic Delta-Reachability: For our purposes we need to consider *probabilistic bounded* reachability: what is the *probability that a SPHS* (which models system *and* controller) *reaches a goal state in a* **finite** *number of discrete transitions*? Reasoning about reachability in nonlinear hybrid systems entails deciding first-order formulae over the reals. It is well known that such formulae are undecidable when they include, *e.g.*, trigonometric functions. A relaxed notion of satisfiability (δ-satisfiability [16]) can be

utilized to overcome this hurdle, and SMT solvers such as dReal [17] and iSAT-ODE [10] can "δ-decide" a wide variety of real functions, including transcendental functions and solutions of nonlinear ODEs. (Essentially, those tools implement solving procedures that are sound and complete up to a given arbitrary precision.)

A probabilistic extension of bounded reachability in SPHSs was presented in [31], which basically boils down to measuring the *goal set*, *i.e.* the set of parameter points for which the system satisfies the reachability property. Recall that the set of goal states for a SPHS is described by its goal predicate. When nondeterministic parameters are present, the system may exhibit a range of reachability probabilities, depending on the value of the nondeterministic parameters. That is, the reachability probability is given by a function $\mathbf{Pr}(\nu) = \int_{G(\nu)} d\mathbb{P}$, defined for any $\nu \in P_N$, where $G(\nu)$ is the goal set and $\mathbb{P}$ is the probability measure of the random parameters. The ProbReach tool utilizes the notion of δ-satisfiability when computing the goal set, thereby computing *probabilistic* δ-*reachability* [30]. In particular, ProbReach computes probability *enclosures* for the range of function $\mathbf{Pr}$ over parameter sets $\mathcal{N} \subseteq P_N$, *i.e.*, intervals $[a,b]$ such that

$$\forall \nu \in \mathcal{N} \quad \mathbf{Pr}(\nu) \in [a,b] \tag{2.1}$$

where $0 \leqslant a \leqslant b \leqslant 1$ (but $a = b$ can only be achieved in very special cases, of course). To solve our synthesis problems we leverage ProbReach's formal approach and statistical approach for the computation of probability enclosures.

Formal Approach: ProbReach guarantees that the returned enclosures satisfy (2.1) *formally* and *numerically* [30]. In particular, any enclosure either has a desired width $\varepsilon \in \mathbb{Q}^+$, or the size of the corresponding parameter box $\mathcal{N} \subseteq P_N$ is smaller than a given lower limit. The computational complexity of this approach increases exponentially with the number of parameters, so it might not be feasible for large systems.

Statistical Approach: It trades computational complexity with correctness guarantees [31], by solving approximately the problem of finding a value ν^* for the nondeterministic parameters that minimizes (maximizes) the reachability probability $\mathbf{Pr}$:

$$\nu^* \in \arg\min_{\nu \in P_N} \mathbf{Pr}(\nu) \qquad \left(\nu^* \in \arg\max_{\nu \in P_N} \mathbf{Pr}(\nu)\right) . \tag{2.2}$$

ProbReach returns an estimate $\hat{\nu}$ for ν^* and a probability enclosure $[a,b]$ that are *statistically* and *numerically* guaranteed to satisfy:

$$\mathrm{Prob}(\mathbf{Pr}(\hat{\nu}) \in [a,b]) \geqslant c \tag{2.3}$$

where $0 < c < 1$ is an arbitrary confidence parameter. In general, the size of the enclosure $[a,b]$ cannot be arbitrarily chosen due to undecidability reasons, although it may be possible to get tighter enclosures by increasing the numerical precision of δ-reachability. Also, the statistical approach utilizes a Monte Carlo (Cross Entropy) method, so it cannot guarantee that $\hat{\nu}$ is a global optimum, *i.e.*, that satisfies (2.2).

PID control: A PID control law is the sum of three kinds of control actions, *Proportional, Integral and Derivative actions*, each of which depends on the *error value*, *e*, *i.e.* the difference between a target trajectory, or *setpoint sp*, and the measured output of the

system y. At time t, the resulting control law $u(t)$ and error $e(t)$ are given by:

$$u(t) = \underbrace{K_p e(t)}_{\text{P}} + \underbrace{K_i \int_0^t e(\tau)\, d\tau}_{\text{I}} + \underbrace{K_d \dot{e}(t)}_{\text{D}}, \qquad e(t) = sp(t) - y(t) \tag{2.4}$$

where constants K_p, K_i and K_d are called *gains* and fully characterize the PID controller.

The above control law assumes a continuous time domain, which is quite common in the design stage of a PID controller. Alternatively, PID control can be studied over discrete time, where the integral term is replaced by a sum and the derivative by a finite difference. However, the analysis of discrete-time PID controllers is impractical for non-trivial time bounds because they induce a discrete transition for each time step, and thus, they directly affect the unrolling/reachability depth required for the bounded reachability analysis, which is at the core of our synthesis method.

3 PID Control of Hybrid Plants

We formally characterize the system given by the feedback loop between a plant SPHS H and a PID controller, so called *closed-loop system* (see Figure 1). We would like to stress that we support plants specified as hybrid systems, given that a variety of systems naturally exhibit hybrid dynamics (regardless of the controller). For instance, in the artificial pancreas case study of Section 5, discrete modes are used to describe different meals, while the glucose metabolism is captured by a set of ODEs.

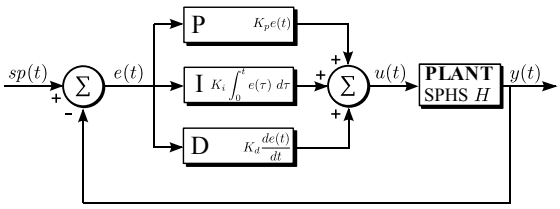


Fig. 1. PID control loop

We assume that the controller is an additive input and can manipulate only one of the state variables of H, x_u, and that for each mode q of H, there is a measurement function h_q that provides the output of the system at q. To enable synthesis, we further assume that the PID controller gains $\mathbf{k} = (K_p, K_i, K_d)$ are (unknown) nondeterministic parameters with domain K. To stress this dependency, below we use the notation $u(\mathbf{k}, t)$ to denote the PID control law of Equation 2.4.

Definition 2 (PID-SPHS control system). *Let $H = \langle Q, \Upsilon, X, P, Y, R, \mathsf{jump}, \mathsf{goal}\rangle$ be a plant SPHS, and let u be a PID controller (2.4) with gain parameters $\mathbf{k} \in K \subset \mathbb{R}^3$. For $q \in Q$, let $h_q : X \to \mathbb{R}$ be the corresponding measurement function. Let x_u be the manipulated state variable, $i_u \in \{1, \ldots, n\}$ be the corresponding index in the state vector, and $sp : [0,t] \to \mathbb{R}$ be the desired setpoint. The* PID-SPHS control system *with plant H is the SPHS $H \parallel u = \langle Q, \Upsilon, X, P \times K, Y', R', \mathsf{jump}, \mathsf{goal}\rangle$, where*

– $Y' = \{\mathbf{y}'_q(\mathbf{p},\mathbf{k},t) : q \in Q, \mathbf{p} \in X \times P, \mathbf{k} \in K, t \in [0,1]\}$, *where the continuous dynamics of each state variable with index* $i = 1,\ldots,n$ *is given by*

$$\mathbf{y}'_{q,i}(\mathbf{p},\mathbf{k},t) = \begin{cases} \mathbf{y}_{q,i}(\mathbf{p},t) + u(\mathbf{k},t) & \text{if } i = i_u \\ \mathbf{y}_{q,i}(\mathbf{p},t) & \text{otherwise} \end{cases}$$

where $\mathbf{y}_{q,i}$ *is the corresponding continuous dynamics in the plant SPHS* H*, and* $u(\mathbf{k},t)$ *is the PID law described in (2.4), with error*

$$e(t) = sp(t) - h_q(\mathbf{y}'_q(\mathbf{p},\mathbf{k},t)); \text{ and}$$

– $R' = \{\mathbf{g}'_{(q,q')}(\mathbf{p},\mathbf{k},t) : (q,q') \in \Upsilon, \mathbf{p} \in X \times P, \mathbf{k} \in K, t \in [0,T]\}$, *where* $\mathbf{g}'_{(q,q')}(\mathbf{p},\mathbf{k},t) = \mathbf{g}_{(q,q')}(\mathbf{p},t)$*, i.e. the reset* $\mathbf{g}'_{(q,q')}$ *is not affected by the controller parameters* $\mathbf{k}$ *and is equal to the corresponding reset of the plant* H*,* $\mathbf{g}_{(q,q')}$*.*

In other words, the PID-SPHS control system is obtained by applying the same PID controller to the continuous dynamics of each discrete mode of the hybrid plant, meaning that the PID-SPHS composition produces the same number of modes of the plant SPHS. We remark that external disturbances as well as plant uncertainties can be encoded through appropriate random variables in the plant SPHS.

4 Safe and Robust PID Controller Synthesis

In this section we first illustrate the class of synthesis properties of interest, able to capture relevant safety and performance objectives. Second, we formulate the PID control synthesis problem and the related problem of maximum disturbance synthesis.

We remark that our main objective is designing PID controllers with formal **safety guarantees**, *i.e.* a given set of bad states should never be reached by the system, or reached with very small probability. Similarly, we aim to synthesize controllers able to guarantee, by design, prescribed performance levels. For instance, the designer might need to keep the settling time within strict bounds, or avoid large overshoot.

To this purpose, we consider two well-established performance measures, the fundamental index (FI) and the weighted fundamental index (FI_w) [24,25][5], defined by:

$$FI(t) = \int_0^t (e(\tau))^2 d\tau \qquad FI_w(t) = \int_0^t \tau^2 \cdot (e(\tau))^2 d\tau. \tag{4.5}$$

FI and FI_w quantify the cumulative error between output and set-point, thus providing a measure of how much the system deviates from the desired behavior. Crucially, they also indirectly capture key transient response measures such as steady-state error, *i.e.* the value of $e(t)$ when $t \to \infty$, or maximum overshoot, *i.e.* the highest deviation from the setpoint[6]. In fact, small FI values typically indicate good transient response (*e.g.* small

[5] FI and FI_w are also also known as "integral of square error" and "integral of square time weighted square error", respectively.

[6] In PID theory, transient response measures are often evaluated after applying a step function to the set-point. However, we do not restrict ourselves to this scenario.

overshoot or short rise-time), while FI_w weighs errors with the corresponding time, in this way stressing steady state errors.

We now formulate the main reachability property for the synthesis of safe and robust controllers, which is expressed by predicate goal. The property captures the set of bad states that the controller should avoid (predicate bad) as well as performance constraints through upper bounds $FI^{\max}, FI_w^{\max} \in \mathbb{R}^+$ on the allowed values of FI and FI_w, respectively, and is given by:

$$\mathsf{goal} = \mathsf{bad} \vee (FI > FI^{\max}) \vee (FI_w > FI_w^{\max}). \tag{4.6}$$

In the case that the designer is not interested in constraining FI or FI_w, we allow $FI^{\max}$ and $FI_w^{\max}$ to be set $+\infty$.

We now introduce the PID controller synthesis problem that aims at synthesizing the control parameters yielding the minimal probability of reaching the goal (*i.e.* the undesired states). Importantly, this corresponds to minimizing the effects on the plant of random disturbances, that is, to *maximizing the robustness* of the resulting system.

We remark that the unrolling depth and the goal predicate are implicit in the reachability probability function **Pr** (see Section 2).

Problem 1 (PID controller synthesis). Given a PID-SPHS control system $H \parallel u$ with unknown control parameters $\mathbf{k} \in K$, find the parameters $\mathbf{k}^*$ that minimize the probability of reaching the goal:

$$\mathbf{k}^* \in \arg\min_{\mathbf{k} \in K} \mathbf{Pr}(\mathbf{k}).$$

For the duality between safety and reachability, Problem 1 is equivalent to synthesizing controllers that maximize the probability that $\neg$goal always holds. If $H \parallel u$ has no random parameters (but only nondeterministic parameters), then Problem 1 is equivalent to synthesizing, if it exists, a controller that makes goal unsatisfiable.

As previously explained, the control parameters $\mathbf{k}$ that we aim to synthesize must be defined as nondeterministic parameters in the SPHS $H \parallel u$. Crucially, we can employ both the formal and the statistical approach alike to solve this problem.

In general, it is not possible to know the exact minimizing parameter because of the inherent undecidability. However, using the formal approach one could select the synthesized controller parameter $\mathbf{k}^*$ *as the midpoint of the parameter box whose enclosure has the least midpoint*. Through the following proposition, we show that this solution can be made arbitrarily precise when all of the returned enclosures have length $\leq \varepsilon$, the user-defined parameter that determines the desired length of the enclosure as explained in Section 2 (however, this cannot be always guaranteed).

Proposition 1. *Suppose that the returned enclosures by the formal approach have all length* $\leq \varepsilon$. *Let* P^* *be the actual minimal probability, and let* $\mathbf{k}^*$ *be the solution of the formal approach for Problem 1. Then, it holds that*

$$\mathbf{Pr}(\mathbf{k}^*) < P^* + \frac{3}{2}\varepsilon\,.$$

Proof. See Appendix A.

On the other hand, the statistical algorithm returns an over-approximation $\hat{P}$ of the minimum probability, c-confidence interval $[\hat{P}]$ such that $\hat{P} \in [\hat{P}]$, and synthesized parameters $\mathbf{k}^*$ whose reachability probability is included in $[\hat{P}]$ with probability at least c, as per Equations 2.2 and 2.3.

Below, we define the maximum disturbance synthesis problem, aimed at finding, given a concrete controller, the maximum disturbance value that the resulting control system can support without violating a given property. This problem is complementary to the PID synthesis problem, since it allows the designer to formally evaluate the robustness of a known controller, possibly synthesized in a previous step. Specifically, we assume that the disturbance is represented by a vector of nondeterministic parameters $\mathbf{d}$ in the plant SPHS, and that $\mathbf{d}$ ranges over some bounded domain D.

Problem 2 (Maximum disturbance synthesis). Given a PID-SPHS control system $H \parallel u$ with *known* control parameters $\mathbf{k}^* \in K$ and *unknown* disturbance $\mathbf{d} \in D$, and a probability threshold p, find the highest disturbance $\mathbf{d}^*$ for which the probability of reaching the goal does not exceed p, *i.e.* such that:

$$\mathbf{d}^* = \max \{\mathbf{d} \in D \mid \mathbf{Pr}(\mathbf{d}) \leq p\}.$$

For the duality between safety and reachability, the probability of reaching goal is below p if and only if the probability that $\neg$goal always holds is above $1-p$. If $H \parallel u$ has no random parameters (but only nondeterministic parameters), then Problem 2 reduces to finding the largest disturbance for which the PID-SPHS system either reaches or does not reach the goal.

Note that the maximum disturbance synthesis problem is fundamentally different from the controller synthesis problem, because the kind of parameters that we seek to synthesize represent external factors that cannot be controlled. That is why we are interested in knowing the maximum (worst-case) value they can attain such that the requirements are met with given probability constraints. In particular, we restrict to upper-bound constraints because we want to limit the probability of reaching a given goal (undesired) state, even though lower bound constraints can be equally supported by the synthesis method.

Problem 2 is solved through the formal approach, which allows identifying the parameters boxes whose probability enclosures are guaranteed to be below the threshold p, *i.e.*, they are intervals of the form $[P_{\min}, P_{\max}]$ with $P_{\max} \leq p$. Then, the synthesized parameter $\mathbf{d}^*$ is selected as the highest value among all such parameter boxes.

It follows that the returned $\mathbf{d}^*$ is guaranteed to meet the probability constraint ($\mathbf{Pr}(\mathbf{d}^*) \leq p$), but, due to the iterative refinement, $\mathbf{d}^*$ under-estimates the actual maximum disturbance. In this sense, $\mathbf{d}^*$ is a safe under-approximation. The reason is that there might exist some "spurious" parameter boxes $[\mathbf{d}]$ (not returned by the algorithm), *i.e.* such that p lies within the corresponding probability enclosure $[P]$ and $[\mathbf{d}]$ contains a disturbance value $\mathbf{d}'$ that is higher than the synthesized $\mathbf{d}^*$ and that, at the same time, meets the constraint $\mathbf{Pr}(\mathbf{d}') \leq p$.

The statistical approach cannot be applied in this case, because it relies on the Cross Entropy method, which is designed for estimation and optimization purposes and is not suitable for decision problems. Note indeed that the probability bound $\leq p$ induces a Boolean (and not quantitative) property.

5 Case Study: Artificial Pancreas

We evaluate our method on the closed-loop control of insulin treatment for Type 1 diabetes (T1D), also known as the *artificial pancreas (AP)* [20]. Together with model predictive control, PID is the main control technique for the AP [32,22], and is found as well in commercial devices [23].

The main requirement for the AP is to keep blood glucose (BG) levels within tight, healthy ranges, typically between 70-180 mg/dL, in order to avoid *hyperglycemia* (BG above the healthy range) and *hypoglycemia* (BG below the healthy range). While some temporary, postprandial hyperglycemia is typically admissible, hypoglycemia leads to severe health consequences, and thus, it should be avoided as much as possible. This is a crucial safety requirement, which we will incorporate in our synthesis properties.

The AP consists of a continuous glucose monitor that provides glucose measurements to a control algorithm regulating the amount of insulin injected by the insulin pump. The pump administers both *basal insulin*, a low and continuous dose that covers insulin needs outside meals, and *bolus insulin*, a single high dose for covering meals.

Meals represent indeed the major disturbance in insulin control, which is why state-of-the-art commercial systems[7] can only regulate basal insulin and still require explicit meal announcements by the patient for bolus insulin. To this purpose, robust control methods have been investigated [28,34,27], since they are able to minimize the impact of input disturbances (in our case, meals) on the plant (the patient). Thus, they have the potential to provide full closed-loop control of bolus insulin without manual dosing by the patient, which is inherently error-prone and hence, dangerous. Our method for the synthesis of safe and robust controllers is therefore particularly meaningful in this case.

5.1 Plant Model

To model the continuous system's dynamics (*e.g.*, glucose and insulin concentrations), we consider the well-established nonlinear model of Hovorka *et al*. [21].

At time t, the input to the system is the infusion rate of bolus insulin, $u(t)$, which is computed by the PID controller. The system output $y(t)$ is given by state variable $Q_1(t)$ (mmol), describing the amount of BG in the accessible compartment, *i.e.* where measurements are taken, for instance using finger-stick blood samples. For simplicity, we did not include a model of the continuous glucose monitor (see *e.g.* [35]) that instead measures glucose in the tissue fluid, but we assume continuous access to blood sugar values. The state-space representation of the system is as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{F}\left(\mathbf{x}(t), u(t), D_G\right), \qquad y(t) = Q_1(t) \tag{5.7}$$

where $\mathbf{x}$ is the 8-dimensional state vector that evolves according to the nonlinear ODE system $\mathbf{F}$ (see Appendix B for the full set of equations and parameters). The model assumes a single meal starting at time 0 and consisting of an amount D_G of ingested carbohydrates. Therefore, parameter D_G represents our input disturbance.

[7] MINIMED 670G by Medtronic `https://www.medtronicdiabetes.com/products/minimed-670g-insulin-pump-system`

Instead of the BG mass $Q_1(t)$, in the discussion of the results we will mainly evaluate the BG concentration $G(t) = Q_1(t)/V_G$, where V_G is the BG distribution volume.

The error function of the PID controller is defined as $e(t) = sp - Q_1(t)$ with the constant set point sp corresponding to a BG concentration of 110 mg/dL. Multiple meals can be modeled through a stochastic parametric hybrid system with one mode for each meal. In particular, we consider a one-day scenario consisting of three random meals (breakfast, lunch and dinner), resulting in the SPHS of Figure 2.

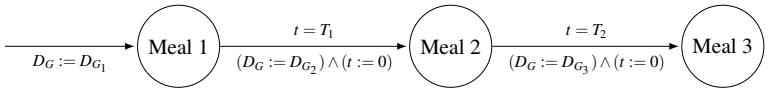


Fig. 2. Stochastic parametric hybrid system modelling a scenario of 3 meals over 24 hours. Above each edge, we report the corresponding jump conditions, below, the resets.

The model features five random, normally-distributed parameters: the amount of carbohydrates of each meal, $D_{G_1} \sim \mathcal{N}(40,10)$, $D_{G_2} \sim \mathcal{N}(90,10)$ and $D_{G_3} \sim \mathcal{N}(60,10)$, and the waiting times between meals, $T_1 \sim \mathcal{N}(300,10)$ and $T_2 \sim \mathcal{N}(300,10)$.

A meal containing D_{G_1} grams of carbohydrates is consumed at time 0. When the time in the first mode reaches T_1 minutes the system makes a transition to the next mode *Meal* 2 where the value of the variable D_G is set to D_{G_2} and the time is reset to 0. Similarly, the system transitions from mode *Meal* 2 to *Meal* 3, resetting variables D_G and t to D_{G_3} and 0, respectively. All remaining variables are not reset at discrete transitions.

Basal insulin and initial state: The total insulin infusion rate is given by $u(t)+u_b$ where $u(t)$ is the dose computed by the PID controller, and u_b is the basal insulin. As typically done, the value of u_b is chosen in order to guarantee a steady-state BG value of $Q_1 = sp$, and the steady state thus obtained is used as the initial state of the system.

We denote with C_0 the basal controller that switches off the PID controller and applies only u_b (*i.e.*, K_p, K_i and K_d are equal to 0).

5.2 Experiments

We apply the formal and statistical techniques of ProbReach to synthesize the controller parameters K_p, K_d and K_i (Problem 1) and the maximum safe disturbance D_G (Problem 2), considering the probabilistic reachability property of Section 4. All experiments in this section were conducted on a 32-core (Intel Xeon 2.90GHz) Ubuntu 16.04 machine, and the obtained results for the synthesized controllers are summarized in Table 1. We also validate and assess performance of the controllers over multiple random instantiations of the meals, which is reported in Figure 3.

PID controller synthesis Typical healthy glucose levels vary between 4 and 10 mmol/L. Since avoiding hypoglycemia ($G(t) < 4$ mmol/L) is the main safety requirement of the artificial pancreas, while (temporary) hyperglycemia can be tolerated and is inescapable

after meals, we will consider a BG range of $[4,16]$ for our safety properties. In this way we protect against both hypoglycemia and very severe levels of hyperglycemia.

Given that the basal insulin level is insufficient to cover meal disturbances, the basal controller C_0 prevents hypoglycemia but causes severe hyperglycemia when a large meal is consumed ($D_G > 80$) or when the BG level is not low enough by the time the next meal is consumed (see Figure 3).

We used the statistical engine of ProbReach to synthesize several controllers (see Table 1), over domains $K_d \in [-10^{-1},0]$, $K_i \in [-10^{-5},0]$ and $K_p \in [-10^{-3},0]$, which minimize the probability of reaching a bad state at any time instant in the modes *Meal* 1, *Meal* 2 and *Meal* 3 (reachability depth of 0, 1 or 2, respectively).

The set of unsafe glucose ranges is captured by predicate $\mathsf{bad} = G(t) \notin [4,16]$. Controller C_1 was synthesized considering only safety requirements, corresponding to the reachability specification $\mathsf{goal} = \mathsf{bad}$ (see Equation 4.6). On the other hand, controllers C_2, C_3 and C_4 were obtained taking into account also performance constraints, by using the default specification (4.6): $\mathsf{goal} = \mathsf{bad} \vee (FI > FI^{\max}) \vee (FI_w > FI_w^{\max})$. Thresholds $FI^{\max}$ and $FI_w^{\max}$ have been set to gradually stricter values, respectively to 3.5×10^6 and 70×10^9 for C_2, 3×10^6 and 50×10^9 for C_3, and 2.7×10^6 and 30×10^9 for C_4.

#	K_d ($\times 10^2$)	K_i ($\times 10^7$)	K_p ($\times 10^4$)	CPU_{syn}	P	CPU_P	$D_{G_1}^{max}$	CPU_{max}
C_0	0	0	0	0	[0.97322,1]	176	69.4	2,327
C_1	-6.02	-3.53	-6.17	92,999	[0.19645,0.24645]	4,937	88.07	3,682
C_2	-5.73	-3.00	-6.39	156,635	[0.31307,0.36307]	64,254	87.62	3,664
C_3	-6.002	-1.17	-6.76	98,647	[0.65141,0.70141]	59,215	88.23	3,881
C_4	-6.24	-7.55	-5.42	123,726	[0.97149,1]	11,336	88.24	3,867

Table 1. Results of PID controller synthesis, where: # – name of the synthesized controller, K_d, K_i and K_p – synthesized values of the gain constants characterizing the corresponding controller (Problem 1), CPU_{syn} – CPU time in seconds for synthesizing the controller parameters, P – 99%-confidence interval for the reachability probability, CPU_P – CPU time in seconds for computing P for synthesized controller, $D_{G_1}^{max}$ – synthesized maximum meal disturbance for which the system never reaches the unsafe state, CPU_{max} – CPU time in seconds for obtaining $D_{G_1}^{max}$.

Due to the high computational complexity of the artificial pancreas model, the controller synthesis was performed in two steps. First, the values of K_p, K_i and K_d were synthesized using a coarse precision (*i.e.*, desired width for confidence intervals P) for computing the probability estimates during the nondeterministic parameter search. Second, the confidence intervals for the obtained controllers were computed with a higher precision. The values of CPU_{syn} and CPU_P in Table 1 represent CPU times used for solving these two steps. The high computation times are due to the fact that the solvers incorporated by ProbReach solve ODEs in a guaranteed manner which is, for general Lipschitz-continuous ODEs, a PSPACE-complete problem, and thus, it is the main bottleneck of the implemented algorithms.

Besides C_0 that unsurprisingly yields the highest probability of safety violation (highest P for the reachability probability), results in Table 1 evidence that control-

lers $C_1, \ldots, C_4$ fail to maintain the safe state with increasingly higher probability. As we shall see in more detail later, this behaviour is mostly due to the performance constraints that become harder and harder to satisfy.

Maximum disturbance synthesis We solve Problem 2 for each of the obtained controllers in Table 1. We consider a domain of $[0, 120]$ for the maximum meal disturbance, and apply the formal approach of ProbReach for synthesizing the maximum size $D_{G_1}^{max}$ of the first meal, such that, given any disturbance $D_{G_1} \in [0, D_{G_1}^{max}]$, the system does not reach the unsafe state within 12 hours. Note that this corresponds to setting the probability threshold p of Problem 2 to 0. Since we are interested in just one meal, we consider a reachability depth of 0 (path length of 1) for the bounded reachability property.

The results in Table 1 indicate that applying a PID controller increases the size of the allowed meal from approximately 69g of the basal controller to about 88g, and at the same time, the difference between the synthesized controllers is negligibly small.

Although introducing a controller does not increase the maximum disturbance dramatically with respect to the basal case, a PID control decreases the BG level sufficiently enough so that a subsequent meal of similar size can be consumed without the risk of experiencing severe hyperglycemia. In contrast, C_0 does not bring the glucose level low enough before the following meal.

Note that, being normally distributed with mean 90 g, the second random meal exceeds such obtained maximum disturbances, which explains why the synthesized controllers fail with some probability to avoid unsafe states.

Performance and safety evaluation In this experiment, we evaluate safety and performance of the controllers by simulating 1,000 instantiations of the random meals. Such obtained glucose profiles and statistics are reported in Figure 3. No hypoglycemia episode ($G < 4$) was registered.

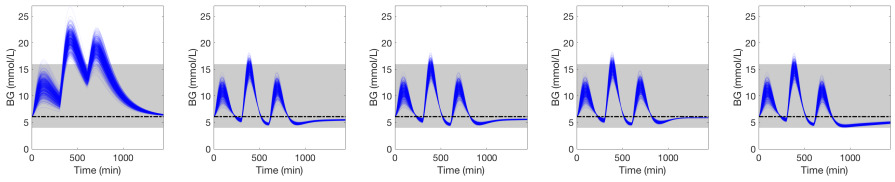
Plots evidence that all four synthesized controllers ($C_1, \ldots, C_4$) perform dramatically better than the basal controller C_0, which stays, on the average, 23.59% of the time in severe hyperglycemia (see index t_{bad}). In particular, all the traces simulated for C_0 violate the safe BG constraints $G \in [4, 16]$ (100% value of $\%_{\mathsf{bad}}$).

On the other hand, controllers $C_1, \ldots, C_4$ violate safe BG constraints for 17-22% of their traces, but this happens only for a very short while (no more than 0.45% of the time) after the second (the largest) meal. This comes with no surprise since we already formally proven that the second meal exceeds the allowed maximum meal disturbance.

C_0 has the worst performance in terms of FI and FI_w, with mean FI and FI_w values (indices $\overline{FI}$ and $\overline{FI_w}$, resp.) significantly larger than those of $C_1, \ldots, C_4$. Among the synthesized controllers, C_3 has the best steady-state behavior (as visible in Figure 3, plot d), keeping the glucose level very close to the set point towards the end of the simulation. C_3 yields indeed the best mean FI_w value (index $\overline{FI_w}$), while the worse steady-state behavior is observed for C_4. On the other hand, mean FI values are very similar, meaning that $C_1, \ldots, C_4$ maintain the BG levels equally far from the set point on the average.

One would expect C_4 to have the best performance in terms of FI_w, since it was synthesized with the stricter FI_w constraint ($FI_w^{\max} = 30 \times 10^9$). This constraint is, however,

too strong to be satisfied, as demonstrated by the 100% value of index $\%_{FI_w > FI_w^{\max}}$ (see Figure 3), implying all traces fail to satisfy $FI_w \leq FI_w^{\max}$. In general, we observe that strengthening the performance constraints leads to higher chances of violating them (see the last three indices of Figure 3). We conclude that performance constraints (and their violation) largely contribute to the reachability probabilities computed by ProbReach (see Table 1) for C_2, C_3 and C_4, whose traces violate FI or FI_w constraints for 28%, 67%, and 100% of the times, respectively.



(a) C_0 (b) C_1 (c) C_2 (d) C_3 (e) C_4

	t_{bad}	$\%_{\mathsf{bad}}$	$\overline{FI}$ ($\times 10^{-6}$)	$\overline{FI_w}$ ($\times 10^{-9}$)	$\%_{FI>FI^{\max}}$	$\%_{FI_w>FI_w^{\max}}$	$\%_{FI>FI^{\max} \vee FI_w>FI_w^{\max}}$
C_0	23.59%	100%	20.27	653.89	NA	NA	NA
C_1	0.45%	22%	**3.21**	66.32	NA	NA	NA
C_2	0.45%	21.4%	3.21	60.91	**28.5%**	**14%**	**28.5%**
C_3	0.51%	24.2%	3.24	**44.93**	67.2%	21.7%	67.2%
C_4	**0.35%**	**17.3%**	3.21	129.05	86.5%	100%	100%

Fig. 3. BG profiles simulated for 1,000 random meals (shaded blue lines). Grey areas indicate healthy BG ranges ($G \in [4,16]$). Dashed black lines indicate the ideal setpoint. t_{bad}: mean proportion of time where $G \notin [4,16]$ (all traces yielded $G > 4$, *i.e.* no hypoglycemia). $\%_{\mathsf{bad}}$: proportion of traces violating $G \in [4,16]$. $\overline{FI}$ and $\overline{FI_w}$: mean FI and FI_w, resp. $\%_{FI>FI^{\max}}$, $\%_{FI_w>FI_w^{\max}}$ and $\%_{FI>FI^{\max} \vee FI_w>FI_w^{\max}}$: proportion of traces violating, resp., either and both performance constraints. The best value for each index is highlighted in bold.

6 Related Work

A number of approaches have been proposed for the PID control of nonlinear and stochastic systems. Among these, nonlinear PID control [33] defines the controller gains as nonlinear functions of the system state, even though performance guarantees have been established only for subclasses of nonlinear systems. Adaptive PID (APID) control [13] supports nonlinear plants with partly unknown dynamics, but no requirements can be guaranteed by design since the unknown dynamics is estimated via sampling the plant output. In contrast, we can synthesize controllers with guaranteed performance for a large class of nonlinear systems (Lipschitz-continuous) while retaining the complete system dynamics. This allows for a fully model-based approach to controller synthesis, which is key in safety-critical applications, where, on the contrary, the model-free online tuning of APID is potentially dangerous.

PID control for Markov jump systems, *i.e.* where the plant is a linear system with stochastic coefficients, is solved as a convex optimization problem in [18,19], while in [9], robust PID control for stochastic systems is reduced to a constrained nonlinear optimization problem. Compared to these approaches, we support models where stochasticity is restricted to random (both discrete and continuous) parameters, with non-deterministic (*i.e.*, arbitrary) parameters and much richer nonlinear dynamics. Another key strength of our method with respect to the above techniques is that design specifications are given in terms of probabilistic reachability properties. These provide rigor and superior expressiveness and can encode common performance indices for PID controllers [25], as shown in Section 4.

Other related work includes the Simplex architecture [29] where, whenever the plant is at risk of entering an unsafe state, the system switches from a high-performance advanced controller to a pre-certified (safe) baseline controller (with worse performance), leading to a potential trade-off between safety and performance. In our approach, performance and safety are instead equal cohorts in the synthesis process. Unlike Simplex, in the *Control Barrier Function* (CBF) approach [3], there is no baseline controller to fall back on: a CBF minimally perturbs a (possibly erroneous) control input to the plant so the plant remains in the safe region. As far as we know, neither Simplex nor CBFs have been designed with a stochastic plant model in mind.

The controller synthesis problem under safety constraints (bounded STL properties in this case) is also considered in [12]. The main differences between this approach and ours is that they focus on Model Predictive rather than PID control, and their system model does not support stochastic parameters. There are a number of formal approaches (*e.g.*, [1]) to control synthesis that consider the sample-and-hold schema typical of discrete-time controllers, but they do not yield PID controllers and cannot handle stochastic hybrid systems. Verification of hybrid control systems with non-deterministic disturbances is considered in [26] and solved through a combination of explicit model checking and simulation. However, unlike our method, it does not support controller synthesis and arbitrary probability distributions for the disturbances.

There has been a sizable amount of work on tools for formal analysis of probabilistic reachability, although they all have limitations that make them unsuitable for our approach. SiSAT [15] uses an SMT approach for probabilistic hybrid systems with discrete nondeterminism, while continuous nondeterminism is handled via Monte Carlo techniques only [11]; UPPAAL [7] uses statistical model checking to analyze nonlinear stochastic hybrid automata; ProHVer [36] computes upper bounds for maximal reachability probabilities, but continuous random parameters are analyzed via discrete over-approximations [14]; U-Check [5] enables parameter synthesis and statistical model checking of stochastic hybrid systems [4]). However, this approach is based on Gaussian process emulation and optimisation, and provides only statistical guarantees and requires certain smoothness conditions on the satisfaction probability function.

Other approaches to solving SMT problems over nonlinear real arithmetic include the complete (over polynomials), yet computationally expensive, cylindrical algebraic decomposition method implemented in solvers like Z3 [8], as well as a recent method [6] based on the incremental linearization of nonlinear functions. However, none of these support ODEs and transcendental functions.

7 Conclusions and Future Work

The design of PID controllers for complex, safety-critical cyber-physical systems is challenging due to the hybrid, stochastic, and nonlinear dynamics they exhibit. Motivated by the need for high-assurance design techniques in this context, in this paper we presented a new method for the automated synthesis of PID controllers for stochastic hybrid systems from probabilistic reachability specifications. In particular, our approach can provide rigorous guarantees of safety and robustness for the resulting closed-loop system, while ensuring prescribed performance levels for the controller. We demonstrated the effectiveness of our approach on an artificial pancreas case study, for which safety and robustness guarantees are paramount.

As future work, we plan to study more advanced variants of the PID design such as nonlinear PID controllers, as well as investigate how common PID tuning heuristics can be integrated in our automated approach to speed up the search for suitable controllers.

Acknowledgements: Research supported in part by EPSRC (UK) grant EP/N031962/1, FWF (Austria) S 11405-N23 (RiSE/SHiNE), AFOSR Grant FA9550-14-1-0261 and NSF Grants IIS-1447549, CNS-1446832, CNS-1445770, CNS-1445770, CNS-1553273, CNS-1536086, CNS 1463722, and IIS-1460370.

References

1. V. Alimguzhin, F. Mari, I. Melatti, I. Salvo, and E. Tronci. Linearising discrete time hybrid systems. *IEEE Transactions on Automatic Control*, PP(99):1–1, 2017.
2. R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, volume 736 of *LNCS*, pages 209–229, 1992.
3. A. D. Ames and J. Holley. Quadratic program based nonlinear embedded control of series elastic actuators. In *CDC*, pages 6291–6298. IEEE, 2014.
4. E. Bartocci, L. Bortolussi, L. Nenzi, and G. Sanguinetti. System design of stochastic models using robustness of temporal properties. *Theor. Comput. Sci.*, 587:3–25, 2015.
5. L. Bortolussi, D. Milios, and G. Sanguinetti. U-check: Model checking and parameter synthesis under uncertainty. In *QEST*, volume 9259 of *LNCS*, pages 89–104, 2015.
6. A. Cimatti, A. Griggio, A. Irfan, M. Roveri, and R. Sebastiani. Invariant checking of NRA transition systems via incremental reduction to LRA with EUF. In *TACAS*, volume 10205 of *LNCS*, pages 58–75, 2017.
7. A. David, K. Larsen, A. Legay, M. Mikučionis, and D. B. Poulsen. UPPAAL SMC tutorial. *International Journal on Software Tools for Technology Transfer*, 17(4):397–415, 2015.
8. L. De Moura and N. Bjørner. Z3: An efficient SMT solver. In *TACAS*, volume 4963 of *LNCS*, pages 337–340, 2008.
9. P. L. T. Duong and M. Lee. Robust PID controller design for processes with stochastic parametric uncertainties. *Journal of Process Control*, 22(9):1559–1566, 2012.
10. A. Eggers, M. Fränzle, and C. Herde. SAT modulo ODE: A direct SAT approach to hybrid systems. In *ATVA*, pages 171–185, 2008.
11. C. Ellen, S. Gerwinn, and M. Fränzle. Statistical model checking for stochastic hybrid systems involving nondeterminism over continuous domains. *International Journal on Software Tools for Technology Transfer*, 17(4):485–504, 2015.
12. S. S. Farahani, V. Raman, and R. M. Murray. Robust model predictive control for signal temporal logic synthesis. In *ADHS*, 2015.

13. M. Fliess and C. Join. Model-free control. *International Journal of Control*, 86(12):2228–2252, 2013.
14. M. Fränzle, E. M. Hahn, H. Hermanns, N. Wolovick, and L. Zhang. Measurability and safety verification for stochastic hybrid systems. In *HSCC*, pages 43–52, 2011.
15. M. Fränzle, T. Teige, and A. Eggers. Engineering constraint solvers for automatic analysis of probabilistic hybrid automata. *J. Log. Algebr. Program.*, 79(7):436–466, 2010.
16. S. Gao, J. Avigad, and E. M. Clarke. Delta-decidability over the reals. In *LICS*, pages 305–314, 2012.
17. S. Gao, S. Kong, and E. M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. In *CADE-24*, volume 7898 of *LNCS*, pages 208–214, 2013.
18. L. Guo and H. Wang. PID controller design for output PDFs of stochastic systems using linear matrix inequalities. *IEEE T. Sys, Man, and Cyb., Part B (Cyb.)*, 35(1):65–71, 2005.
19. S. He and F. Liu. Robust stabilization of stochastic markovian jumping systems via proportional-integral control. *Signal Processing*, 91(11):2478–2486, 2011.
20. R. Hovorka. Closed-loop insulin delivery: from bench to clinical practice. *Nature Reviews Endocrinology*, 7(7):385–395, 2011.
21. R. Hovorka et al. Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. *Physiological Measurement*, 25(4):905, 2004.
22. L. M. Huyett et al. Design and evaluation of a robust PID controller for a fully implantable artificial pancreas. *Industrial & Engineering Chemistry Research*, 54(42):10311–10321, 2015.
23. S. S. Kanderian Jr and G. M. Steil. Apparatus and method for controlling insulin infusion with state variable feedback, July 15 2014. US Patent 8,777,924.
24. W. S. Levine. *The control handbook*. CRC Press, 1996.
25. Y. Li, K. H. Ang, G. C. Chong, W. Feng, K. C. Tan, and H. Kashiwagi. CAutoCSD-evolutionary search and optimisation enabled computer automated control system design. *International Journal of Automation and Computing*, 1(1):76–88, 2004.
26. T. Mancini, F. Mari, A. Massini, I. Melatti, F. Merli, and E. Tronci. System level formal verification via model checking driven simulation. In *CAV*, volume 8044 of *LNCS*, pages 296–312, 2013.
27. N. Paoletti, K. S. Liu, S. A. Smolka, and S. Lin. Data-driven robust control for type 1 diabetes under meal and exercise uncertainties. In *CMSB, accepted*, 2017.
28. R. S. Parker, F. J. Doyle, J. H. Ward, and N. A. Peppas. Robust H_∞ glucose control in diabetes using a physiological model. *AIChE Journal*, 46(12):2537–2549, 2000.
29. L. Sha. Using simplicity to control complexity. *IEEE Software*, 18(4):20–28, 2001.
30. F. Shmarov and P. Zuliani. ProbReach: Verified probabilistic δ-reachability for stochastic hybrid systems. In *HSCC*, pages 134–139. ACM, 2015.
31. F. Shmarov and P. Zuliani. Probabilistic hybrid systems verification via SMT and Monte Carlo techniques. In *HVC*, volume 10028 of *LNCS*, pages 152–168, 2016.
32. G. M. Steil et al. The effect of insulin feedback on closed loop glucose control. *The Journal of Clinical Endocrinology & Metabolism*, 96(5):1402–1408, 2011.
33. Y. Su, D. Sun, and B. Duan. Design of an enhanced nonlinear PID controller. *Mechatronics*, 15(8):1005–1024, 2005.
34. P. Szalay, G. Eigner, and L. A. Kovács. Linear matrix inequality-based robust controller design for type-1 diabetes model. *IFAC Proceedings Volumes*, 47(3):9247–9252, 2014.
35. M. E. Wilinska et al. Simulation environment to evaluate closed-loop insulin delivery systems in type 1 diabetes. *Journal of diabetes science and technology*, 4(1):132–144, 2010.
36. L. Zhang, Z. She, S. Ratschan, H. Hermanns, and E. M. Hahn. Safety verification for probabilistic hybrid systems. In *CAV*, volume 6174 of *LNCS*, pages 196–211, 2010.

A Proof of Proposition 1

Proof. Let $[\mathbf{k}_m]$ be the parameter box from which $\mathbf{k}^*$ was selected, and let $[P_m] = [P_m^\perp, P_m^\top]$ be the corresponding probability enclosure with minimal midpoint. In the best case, $[P_m]$ has also the least lower bound, implying that $P^* \in [P_m]$ and in turn that, $\mathbf{Pr}(\mathbf{k}^*) \leq P^* + \varepsilon$. In the worst case, there exists another enclosure, $[P_M] = [P_M^\perp, P_M^\top]$ with a better lower bound than $[P_m]$, i.e, with $P_M^\perp < P_m^\perp$. This implies that the actual minimal probability might be in $[P_M]$ and not in $[P_m]$, which induces a worst-case probability error of $P_m^\perp - P_M^\perp$, leading to $\mathbf{Pr}(\mathbf{k}^*) \leq P^* + \varepsilon + P_m^\perp - P_M^\perp$. Now note that $P_m^\perp - P_M^\perp$ cannot exceed the half length of $[P_M]$, because otherwise $[P_M]$ would be the enclosure with the lowest midpoint. It follows that $\mathbf{Pr}(\mathbf{k}^*) < P^* + \varepsilon + \varepsilon/2$.

B Gluco-regulatory ODE model

$$
\begin{aligned}
&\dot{Q}_1(t) = -F_{01} - x_1 Q_1 + k_{12} Q_2 - F_R + EGP_0(1 - x_3) + 0.18 U_G; \\
&\dot{Q}_2(t) = x_1 Q_1 - (k_{12} + x_2) Q_2; \quad U_G(t) = \frac{D_G A_G}{0.18 t_{maxG}^2} t e^{\frac{-t}{t_{maxG}}}; \\
&G(t) = \frac{Q_1(t)}{V_G}; \; \dot{S_1}(t) = u(t) + u_b - \frac{S_1}{t_{maxI}}; \; \dot{S_2}(t) = \frac{S_1 - S_2}{t_{maxI}}; \\
&\dot{I}(t) = \frac{S_2}{t_{maxI} V_I} - k_e I; \quad \dot{x_i}(t) = -k_{a_i} x_i + k_{b_i} I; \quad (i = 1,2,3)
\end{aligned}
\tag{B.8}
$$

The model consists of three subsystems:

- *Glucose Subsystem*: it tracks the masses of glucose (in mmol) in the accessible ($Q_1(t)$) and non-accessible ($Q_2(t)$) compartments, $G(t)$ (mmol/L) represents the glucose concentration in plasma, EGP_0 (mmol/min) is the endogenous glucose production rate and $U_G(t)$ (mmol/min) defines the glucose absorption rate after consuming D_G grams of carbohydrates. D_G represents the main external disturbance of the system.
- *Insulin Subsystem*: it represents absorption of subcutaneously administered insulin. It is defined by a two-compartment chain, $S_1(t)$ and $S_2(t)$ measured in U (units of insulin), where $u(t)$ (U/min) is the administration of insulin computed by the PID controller, u_b (U/min) is the basal insulin infusion rate and $I(t)$ (U/L) indicates the insulin concentration in plasma.
- *Insulin Action Subsystem*: it models the action of insulin on glucose distribution/transport, $x_1(t)$, glucose disposal, $x_2(t)$, and endogenous glucose production, $x_3(t)$ (unitless).

The model parameters are given in Table 2.

par	value	par	value	par	value
w	100	k_e	0.138	k_{12}	0.066
k_{a_1}	0.006	k_{a_2}	0.06	k_{a_3}	0.03
k_{b_1}	0.0034	k_{b_2}	0.056	k_{b_3}	0.024
t_{maxI}	55	V_I	$0.12 \cdot w$	V_G	$0.16 \cdot w$
F_{01}	$0.0097 \cdot w$	t_{maxG}	40	F_R	0
EGP_0	$0.0161 \cdot w$	A_G	0.8		

Table 2. Parameter values for the glucose-insulin regulatory model. w (kg) is the body weight.